EXHIBIT M

INTERNET ARCHIVE

www.archive.org
415.561.6767
415.840-0391 e-fax

mail:
Internet Archive
PO Box 29244
San Francisco, CA
94129-0244

ship:
Internet Archive
116 Sheridan Avenue
Presidio of San Francisco
San Francisco, CA 94129

AFFIDAVIT OF PAUL HICKMAN

1. I am the Office Manager at the Internet Archive, located at the Presidio of San Francisco, California. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive is affiliated with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to surf more than 55 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can type in a URL (i.e., a website address), select a date range, and then begin surfing on an archived version of the Web. The links on the archived files, when served by the Wayback Machine, point to other archived files (whether HTML pages or images). If a visitor clicks on a link on an archived page, the Wayback Machine will serve the archived file with the closest available date to the originally requested page.

4. The Internet Archive receives data from third parties who compile the data by using software programs known as crawlers that surf the Web and automatically store copies of website files at certain points in time as they existed at that point in time. This data is donated to the Internet Archive, which preserves and provides access to it.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL]. Thus, the Internet Archive URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). Typically, a printout from a Web browser will show the URL in the footer. The date assigned by the Internet Archive applies to the HTML file but not to image files linked therein. Thus images that appear on the printed page may not have been archived on the same date as the HTML file. Likewise, if a website is designed with "frames," the date assigned by the Internet Archive applies to the frameset as a whole, and not the individual pages within each frame.

6. Attached hereto as Exhibit A are true and accurate copies of printouts of the Internet Archive's records of the HTML files archived from the URLs and the dates specified in the footer of the printout.

7. I declare under penalty of perjury that the foregoing is true and correct.

DATE: 4/19/06

Paul Forrest Hickman

Golden State Notary Acknowledgment Form

State of California
County of San Francisco } ss.

On 4/19/06 before me, David Ryan.,
personally appeared Paul Forrest Hickman.

personally known to me (or proved to me on the basis of satisfactory evidence) to be the person(s) whose name(s) is/are subscribed to the within instrument and acknowledged to me that he/she/they executed the same in his/her/their authorized capacity(ies), and that by his/her/their signature(s) on the instrument the person(s), or the entity upon behalf of which the person(s) acted, executed the instrument.

DAVID RYAN
COMM. # 1646323
NOTARY PUBLIC-CALIFORNIA
SAN FRANCISCO COUNTY
My Commission Expires
February 19, 2010

WITNESS my hand and official seal.

Signature of Notary

Affidavit of Paul Hickman

Notes

Paul Hickman.

Please provide information about the document that this form is attached to
***This is not required under California State notary public law ***

www.GoldenStateNotary.com (888) 263-1977

RealSecure 1.2

User Guide and Reference Manual

Chapters:

Appendix:

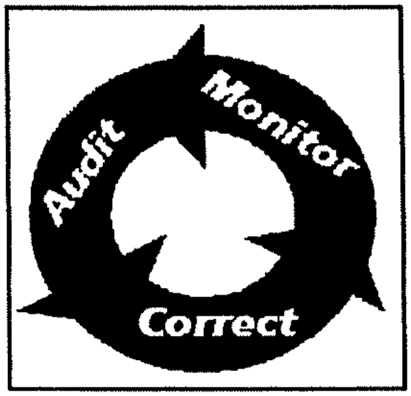
ISS
INTERNET SECURITY SYSTEMS

ISS Technical Support: rs-support@iss.net

06/09/97

Chapter 1: Introduction to Real-Time Intrusion Detection

Security is a hot topic today, especially on the Internet. Very few people understand how to achieve an acceptably secure system. Some companies sell products that promise to make a network secure. What they neglect to mention is... *"there is no single solution to security"*. Security is a process and method of doing business that requires continual updating.



The Security Cycle

➤ *Audit*

The illustration above depicts the Security Cycle used to dynamically improve the security of a system. An auditing tool is used to find the holes in a network. Combining a security audit with a security policy establishes the original baseline security.

➤ *Correct*

The parts of the system which failed the audit are corrected. Subsequently, the system needs to be audited again and again to ensure that new holes do not appear or old ones resurface.

➤ *Monitor*

A monitoring tool is used to watch for new security breaches or attempts to abuse old holes. This keeps the security manager in touch with the state of the network.

RealSecure... falls into the third category. It resides on a computer connected to the network and watches the network traffic. This allows it to compare traffic to a wide variety of attack signatures. It summarizes the information in a concise manner so the security manager can understand what is happening on the network, *as it happens*.

Common Uses for Network Intrusion Detection

RealSecure is very versatile and therefore performs many different functions. Some possible uses include:

➤ **Auditing network utilization....**

- How many hits does each Web server get?
- Who is transferring files from the FTP site?

➤ **Providing a second tier of security behind a firewall....**

- There should never be `Telnet` sessions incoming to the internal network. So, kill and log any such attempts.

➤ **Obtaining profiles of a detected intruder and assessing damage...**

- So, perhaps it is known that the intruder is using the Web server to attack other sites. In this case, consider logging all of his/her session data throughout the night and replay it later.

Remember, **RealSecure** can analyze **any** network traffic. When watching the network for invalid login attempts, **RealSecure** prints all invalid login attempts made over the network. This is a great improvement over normal auditing, in which the administrator must search a log from each system on the network for suspicious activity.

Risks

Because **RealSecure** watches and responds to network events, there are some associated risks. First, it can log all data in a connection including keystrokes and e-mail. Exercise discretion when using this feature. Once logs are created, they should be kept on a secure host as they may contain passwords or other sensitive data. This is one reason to run **RealSecure** on a dedicated machine. Intruders will perceive the monitoring machine as a prime target, both to steal its saved data and to erase evidence of their actions.

Second, **RealSecure** responds to events. In particular, the 'kill' option, if misused, can block traffic over an entire network. Ensure that any connections being killed are ones that require blocking. A wildcard rule with a 'kill' action will block all connections, including `http`, `Telnet`, and `ftp`.

The rule of thumb to use is "think twice, configure once"!

License

Click here to view the **RealSecure** License Agreement.

Legality

In most cases, the United States Government has upheld the right of individuals to monitor their own networks. The general consensus is to notify all users of monitoring. Consult a lawyer if there are any questions as to the legality of this product's use. For explanation of legal issues, refer to the CERT advisory provided herein.

Installing RealSecure

Return to the Index

Chapter 2:
Installing RealSecure™

To obtain maximum benefits, install RealSecure™ on a dedicated machine at an entry point to the network. Good places to consider would be at the Ethernet interface just inside the firewall or between the Internet router and the internal machines. For security and performance, it is strongly recommended that RealSecure be run on its own dedicated machine (ideally, the machine should have as many of its own services as possible disabled). Also, the only user(s) should be the administrator(s). RealSecure collects a lot of data from the network, so the more power and disk space the machine has, the better!

This chapter covers the following topics:

➤ **System Requirements**

➤ **How To Install RealSecure™**

System Requirements

The **RealSecure** engine requires the following configuration:

- SunOS 4.1.x, Solaris 2.3 and up, or Linux (kernel versions 1.3.x and later).
- An Ethernet interface connected to the target network.
- 486-class machine or better; however, for non-dedicated machines running **RealSecure**, more resources will be necessary.
- A minimum of 25 MB of available disk space.
- A minimum of 16 MB RAM, 32 MB recommended.

The **RealSecure** GUI requires the following configuration:

- The X-Window system, version 11 or later.
- Motif Installation (Solaris 2.3 and 2.4 only).
- A minimum of 32 MB RAM.

How to Install RealSecure

Key Processing

To activate **RealSecure** obtain a license key file (`iss.key`). To obtain a license key file, please contact ISS immediately at (770) 395-0150 or by E-mail at keys@iss.net. Upon receiving a license key, save the key as the filename `iss.key` in the /usr/local directory.

Old versions of the software (Release 1.0) will continue to operate using the old key, but to utilize the new functionality of **RealSecure**, obtain a new key from ISS. To obtain a new key, e-mail the existing key along with name and address of the organization and the e-mail address where the new key is to be sent to keys@iss.net. ISS will update key processing records and forward the new key promptly.

Note: Save the e-mail message as "iss.key" in your /usr/local directory It is not necessary to decode this key. Save the entire message or just the key. The key MUST include the "BEGIN" and "END" lines.

For further questions about key processing, contact support@iss.net.

Step 1: Obtain the Distribution Software

Installing from the ISS Web site:

To download **RealSecure** from ISS, access:

http://www.iss.net/RealSecure from within a Web browser.

Step 2: Copying the Distribution Software to the Destination System

The `tar` file must be loaded to each machine that will be running either the **RealSecure** GUI or engine. Methods to transfer the archive include FTP and e-mail.

Perform the remaining steps on *each* machine running **RealSecure**.

Step 3: Untar the Archive and Run the Install Program

Change to the directory the directory where the `tar` file resides, for example:

```
# cd/usr/local
```

Next, enter the following commands:

```
# tar xvf rs-* tar

# cd rs

# ./install.rs
```

Step 4: Start sssd on Each System Running the RealSecure Engine

Note: Perform Step 4 only for multiple engines.

- To run the **RealSecure** engine, install `sssd` on the host.
- To run the **RealSecure** engine remotely, or to run multiple engines in order to sniff multiple segments or networks, perform these tasks for each machine for which the **RealSecure** GUI will run.

Installing from CD-ROM:

Mount the CD-ROM volume (refer to the UNIX manual for instructions).

From the directory where the CD-ROM is mounted, perform: ./install.rs

A. How to Set Up the sssd Engine

The engine and **RealSecure** GUI setup can be automated by running sss-setup. sssd engines require an authentication file be created to restrict access to authorized hosts.

To manually edit the file, perform the following procedure:

- Log in as `root` on the machine on which the engine will run.
- Create a file containing the name or IP address of each machine on the network where the **RealSecure** GUI is being installed. Indicate a random pass phrase used to authenticate the connection.

This authentication file specifies allowable hosts for connectivity to `sssd`, in order to start up engines and to indicate a pass phrase used during connections for authentication and encryption. The default location of this file for the `sssd` server is `/etc/sssd.auth`. To change the default, specify the `-a` option along with an alternate pathname for the authentication file on the command line when running `sssd`.

The format of the authentication file is: <Pass Phrase>//<hostname>

`<hostname>` can be a domain name or IP address. `<Pass Phrase>` should be a unique, hard-to-guess set of letters, numbers and punctuation. Do **not** use a single word, sentence or sentence fragment from a well-known published book or song, or one that is guessable. The `sssd` server runs an engine capable of monitoring all network traffic. Access through the `sssd` server poses serious compromise to the security of a system. Because IP addresses can be spoofed, it is vital to system security to choose a good pass phrase. The authentication file is owned by `root` and must **not** be readable by users on the system other than `root`. Be sure to create the file with proper modes **before** entering pass phrases. Take a look at the this example:

```
# whoami

root

# cd /etc

# echo > sssd.auth

# chmod 600 sssd.auth

# echo ad98lU Aj2 ah c89 kqaknsdh//myguibox.mydomain.com > sssd.auth
```

B. Set Up the GUI

To manually setup the GUI, perform the following procedure:

Run **RealSecure** on the machine from which the **RealSecure** GUI will run. Next, create an authentication file containing the hostname and pass phrases of each host running `sssd` for which **RealSecure** will run. This includes the local host when only planning to run local engines.

The default filename for this file is `/etc/sss.auth`. Match the entries in this file to the pass phrase in the `/etc/sssd.auth` file on each machine. If running only a local engine, accomplish this by simply copying the `sssd.auth` file to `sss.auth`. As with the `sssd.auth` file, be sure no one other than `root` can read the contents of the file.

Because the machine running the **RealSecure** GUI controls the remote engines on all the machines in the `sss.auth` file, it is extremely important the machine remain secure against attack. If at all possible, run the GUI on a machine having no untrusted users and no services running. Take a look at the following example:

```
# whoami

root

# cd /etc
# echo ad98IU Aj2 ah c89 kqaknsdh//ssdbox1.mydomain.com > sss.auth
# echo okPui uz 472 JK cnzx opzutb//sssdbox2.mydomain.com >> sss.auth
```

C. How to Start sssd

To start the `sssd` daemon from a boot startup file, place the following command in a system `rc` file the system uses (i.e.,`/etc/rc`, `/etc/rc.local`, or `/etc/rc.2/SXX.sss`, etc.). The file location depends on the operating system:

Where: `<RealSecure_distrib_directory>` is the directory path in which the **RealSecure** engine is installed.

Start `sssd` from a shell command prompt by entering the same command provided above while logged in as `root`. For verbose output, run the foreground task using the

`-f` option and use the `-v` option along with `-f`.

The following summarizes `sssd` command line options:

`-a <file>`	Authentication file and path (default: `/etc/sssd.auth`)
`-f`	Foreground processing (do not run as daemon)
`-p <port>`	Port number to listen for connections
`-s`	Log connections to `syslog`
`-v`	Verbose output

How to Stop sssd:

Stop the `sssd` daemon with the `kill` command, for example:

```
# kill - TERM 'ps ax | grep sssd | grep -v grep | cut -f1 -d" '"
```

Arguments to `ps` may vary.

Note: It is not necessary to start the `sssd` server if only running an engine on the local machine.

Step 5: How to Start RealSecure GUI (rsgui) on the Administrative Machine

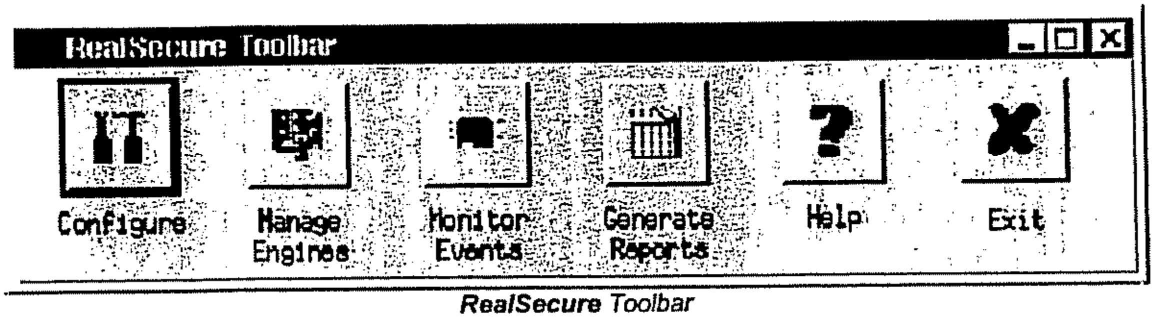
Enter the following commands:

```
# cd rs

# ./rsgui
```

Note: Upon entering the "rsgui" command, an About box displays and will eventually time out if the OK button is not pressed.

The **RealSecure** toolbar displays.



***RealSecure** Toolbar*

The toolbar allows for:

- General Configuration
- Configuring **RealSecure** engines
- Viewing network events
- Generating reports
- Help
- Exit

[Configuring RealSecure]

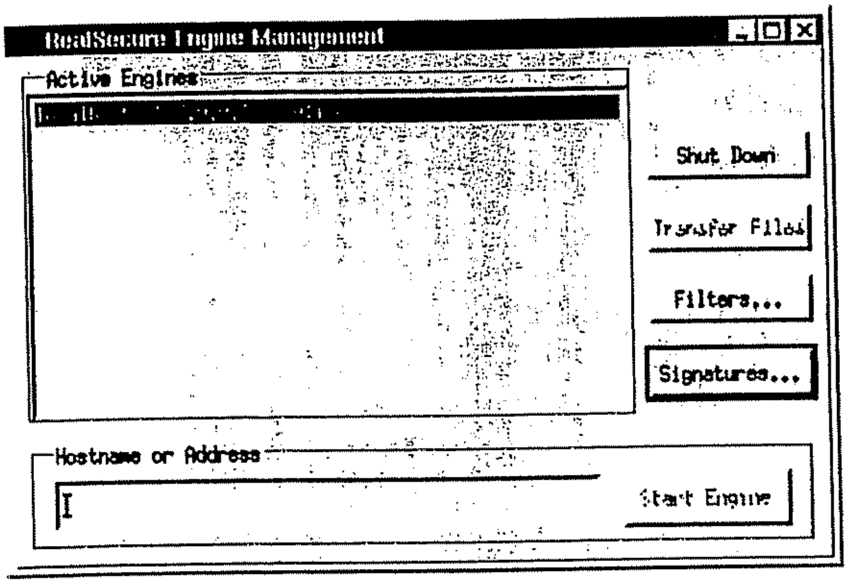
[Return to the Index]

Chapter 3: Configuring RealSecure™

To effectively monitor network problems, provide RealSecure with important network configuration requirements. For instance, a Web server should be having Web traffic, but the company's accounting machine probably should not. In this case, consider configuring RealSecure to ignore Web traffic to the Web server, while continuing to log all other Web traffic. Each service accessed through the network in the same manner.

RealSecure has two related, but separate configuration modes. The filter configuration mode sets what services **RealSecure** watches for connections. The feature configuration mode enables, disables, and fine-tunes custom attack signatures.

To set GUI configurations, access the Engine Management window. To display this window, choose Configure Engines from the **RealSecure** toolbar.

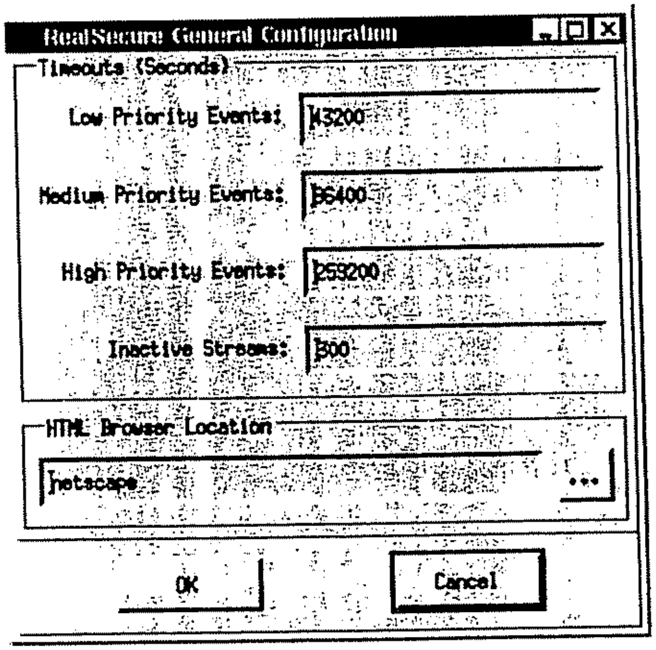


Engine Management window

The **RealSecure** Engine Management screens allow for configuration and monitoring of the state of **RealSecure** engines on the network. The active engines window displays a list of all **RealSecure** engines currently communicating with the network's GUI. Each engine normally remains in an "Alive" state. The GUI periodically pings each engine to check for responsiveness. If it fails to respond to a ping, it moves into a "No Response" state, indicating the engine may no longer be communicating properly with the GUI.

At the bottom of the Engine Management window is a "Hostname or Address" field. Entering the name or IP address of an engine and then pressing Enter or clicking on "Start Engine" causes the GUI to communicate with the "sssd" daemon running on the remote machine; this starts a **RealSecure** engine on that machine. For an engine to startup remotely, **RealSecure** must be installed and the authentication entries should be setup and running sssd on the remote machine. To start up a local engine, use the hostname "localhost" or the IP address "127.0.0.1". Starting up local engines does not require sssd to be running or any authentication information to be configured.

By selecting a running engine, one can then click on any of the buttons (located on the right side of the screen) and perform the specified actions.



Timeouts (Seconds)

Low Priority Events: 43200

Medium Priority Events: 86400

High Priority Events: 259200

Inactive Streams: 300

HTML Browser Location

netscape

General Configuration window

The General Configuration window allows for adjustment of **RealSecure** timeouts and settings to fit a particular network configuration. Once settings are configured, additional changes to this screen are not necessary, unless the system or network configuration has also changed.

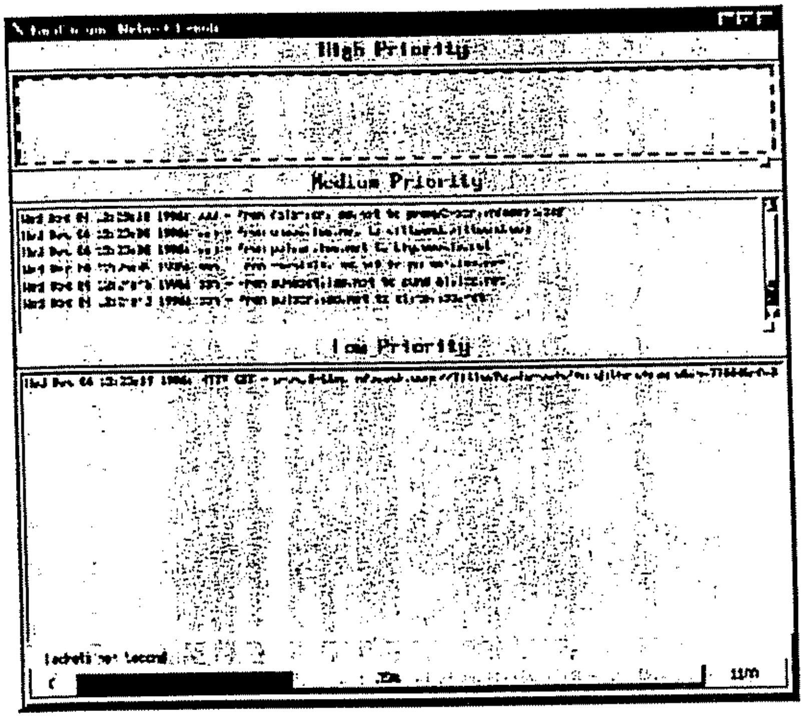
The three event timeout boxes allow for entering the number of seconds that events remain on the GUI screen and is monitored internally by the **RealSecure** engine. The higher these timeout values, the more memory the engine and GUI utilize to track all of the events. Regardless of these values, if logging for events is enabled, a permanent record of each event that occurs in the log files is kept.

The Inactive Streams timeout allows for setting a value for how long the TCP stream data is kept in memory before it is discarded. Decreasing this value will save memory, but if set too low, signatures that occur with a larger amount of time between packets may be interpreted separately instead of as a continuous stream.

The HTML browser location is the path of the Web browser. If a Web browser is in the path, there is no need to specify a full pathname, although it is recommended in case the path variable should change at some point. Clicking on the "..." icon displays a directory browse window from which the Web browser is found.

Filter Configuration

The filter configuration does exactly what its name implies; it selects the types of network connections **RealSecure** should ignore or watch. To edit the filter configuration, click on the desired engine for configuration in the Engine Management window and then click on **Filters**. The Configure Events window displays. Alternately, the `filter.cfg` file on the engine's host are editable.



Configure Events window

A filter consists of rules matching in order. Once a match is found, further searches cancel. If the engine being configured has a filter configuration, the rules appear in the Filter List. To modify a rule, click on the rule. It will appear in the bottom half of the window. Clicking on any of the fields allows for modification. Clicking on the **Modify** button commits changes to the list. To add a rule, click on the position in the list where the new rule is to be inserted. Enter the desired rule in the fields in the bottom half of the window. To add the new rule, click on the **Add** button.

There are several fields to a rule, they are as follows:

- Source and Destination Address -- IPs or ranges of IPs
- Source and Destination Service -- TCP/UDP ports
- Source and Destination Type -- ICMP only
- Protocol -- TCP, UDP, or ICMP
- Label -- a one-word description of the event that appears in the GUI
- Priority -- the severity of this rule
- Actions -- what to do when this rule is matched.

Addresses

The source and destination addresses are structured in the common dotted decimal form (i.e., 10.1.2.3). To specify a range of addresses, use the asterisk (*) wildcard. For instance, an address of 10.1.1.* would match 10.1.1.2 and 10.1.1.50, but not 10.1.18.2. Wildcards must be on even boundaries. For example, 10.* is valid, but 10.1.1* is not. Finally, a wildcard by itself will match all addresses.

Services

Services are the ports in a connection. For instance, HTTP (Web) traffic uses port 80. To select a service, click on the button. A list of services appears. Select the desired service and click OK. Selecting the **Any (0)** service matches any service. If the desired service is not in the list, edit the `services` file included with the distribution software to add the desired service.

ICMP Types

Every ICMP packet has a type and sub-type. For instance, `ping` packets have an ICMP type of Echo Request. To select a type, click on the button. A list of types appears. Select the desired type and click OK. If the desired service is not in the list, edit the `services` file included with the distribution software to add the desired ICMP type.

Protocols

Each rule must be of a specific protocol type. Valid protocols are **TCP**, **UDP** and **ICMP**. TCP is a reliable data transport used for services like E-Mail, FTP, and HTTP. UDP is a datagram service used for services like CU-SeeMe and Talk. Lastly, ICMP is used for sending control messages between Internet nodes.

Labels

The label is a one-word tag for this particular rule. It appears in the logs and on the display. It allows for differentiation of connections at a glance. Valid tags include `Web-Traffic` or `Bob's_PC`, but **not** `My Server` (note the space).

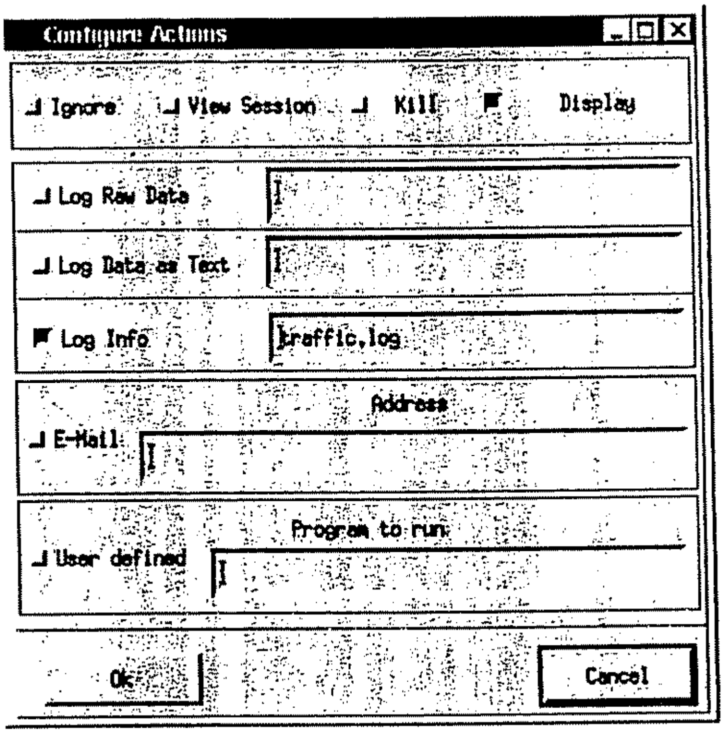
Priority

Each rule has a priority that controls which window the event appears and additional group events for generating reports. Valid priorities are **high**, **medium**, and **low**. It is best to group rules based on priority, to filter out common events (Web transfers, e-mail) from less common events (attempts to exploit security holes, connections to the accounting machine).

Actions

The action configuration is the same for both filters and attack pattern matching. There are quite a few possible actions, they are:

- Ignore any event matching this rule
- Display a message in the main window where the event occurred
- View the data from the connection in real-time
- Kill the connection by sending a reset packet (only possible with TCP connections)
- Mail a notification to the administrator
- Run a User-specified program when the event occurs
- Log data to a file:
  - Log Info that the connection occurred to a file
  - Log Text data sent through the connection
  - Log Raw data sent through the connection for later playback



Configure Actions window

To enable or disable an action, click on it. When the **Ignore** action is enabled, all other actions are disabled. Some actions require additional data. For instance, if the **Mail** action is enabled, **RealSecure** must have an address in order to send mail. After selecting the desired action(s), click OK to continue.

Sample Uses of Filter Rules

Getting an Idea of Your Network Traffic

To initially become accustomed to filters and to become familiar with network traffic patterns, use `filter.cfg` (which is the default).

Be sure to log in as `root`. Follow the startup instructions for using **RealSecure**. This configuration displays all TCP and UDP connections on the network. The display can get very crowded, quickly. Select **Quit**.

Review the network security policy. Is `rlogin` permitted from anywhere, or just from internal hosts? Examining the firewall's configuration is helpful. The entries in the `filter.cfg` file are matched sequentially (one-by-one). A match indicates the action specified at the end of the line is taken, and further matches are discarded. For this reason, save wildcard entries until the end.

Determine the filter rules based on the network security policy. Usually, these correspond with the firewall configuration. Thus, **RealSecure** is used as a second level of defense. **RealSecure** will recognize and display exactly all occurrences of unauthorized traffic.

To see a general configuration that shows all common services refer to the filter.cfg included by default in the rs directory.

Since **RealSecure** engines communicate with the GUI host via normal UDP packets, configure **RealSecure** to ignore them. Use the following rules to ignore all **RealSecure** reports destined for the GUI host:

```
udp 0.0.0.0/0 10.0.0.1/32 0 900 RSgui 3 ignore
udp 0.0.0.0/0 10.0.0.1/32 0 901 RSeng 3 ignore
```

Sample Configurations

To get started using **RealSecure**, three sample configurations are included. To use them, copy the sample `filter.cfg` and `features.cfg` to each of the hosts that will be using them. The following describes each of the three configurations:

- All filter/ features (default) – Enables all **RealSecure** checking and data decoding.
- Exp filter/ features – Enables only exploit checking (leaves out some of the general information features).
- Web filter/ features – Enables Web checks only.

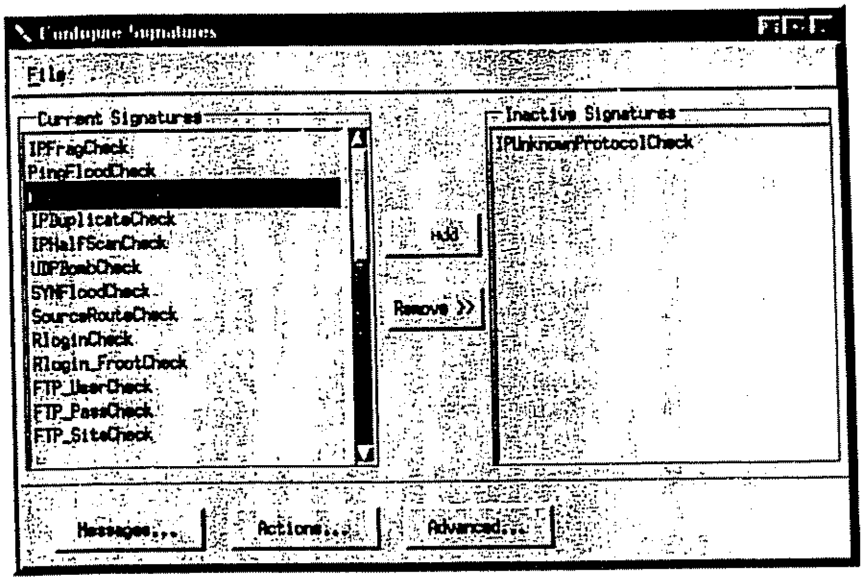
Feature Configuration

RealSecure has a standard set of attack signature checks that examine data inside a connection and check for significant signs of intrusion attempts. For instance, one of the old Sendmail exploits involved passing a malformed **From** address in an e-mail header. **RealSecure** watches all SMTP connections for this invalid data and triggers an alarm once it detects such an attack. The network administrator can then investigate and use the logging and response options of **RealSecure** to trace and ultimately lock out the intruder.

To enable signature checking, the filter configuration must have an entry for the service. For instance, Sendmail bug check is enabled (see above), but port 25 (Sendmail) is not being watched, subsequently the check is never used.

For a list of the checks and a definition of their capabilities, refer to Appendix A, **RealSecure** Features and Attack Signatures.

Each check has a standard set of options, configurable through the GUI. Select the engine to configure in the Engine Management window and then click on the **Signatures** button. The Configure Signatures window displays. Alternately, the `features.cfg` file is editable.



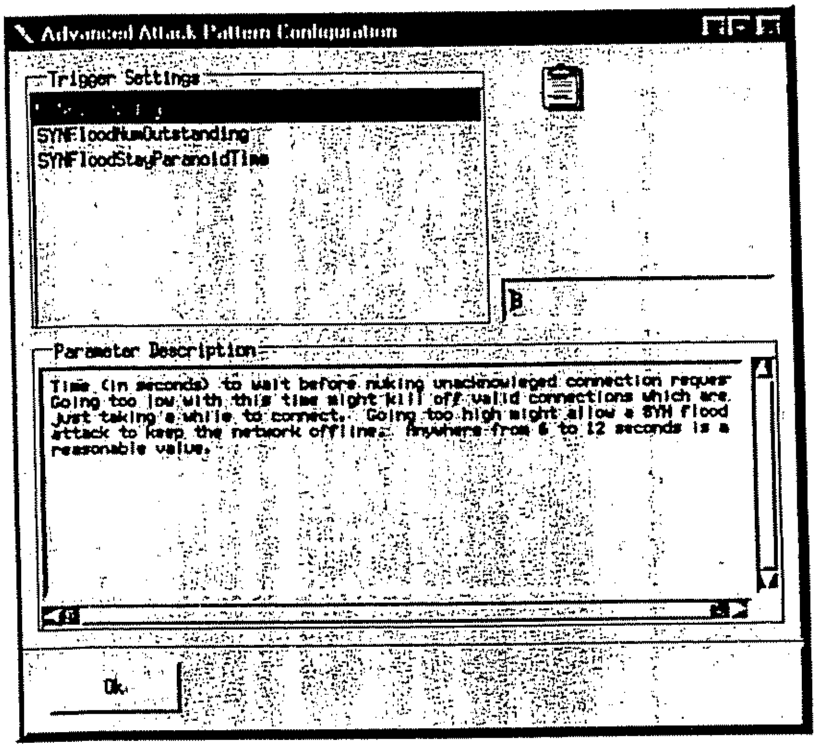
Configure Signatures window

Each check has a standard format scheme for its options. For instance, the check for IP fragmentation attacks contains the prefix **IPFrag**. To manually change the priority of this check, modify the entry called **IPFragPriority** in the `features.cfg` file.

The following details common options for all checks:

- ...Check -- whether to perform this check (yes or no)
- ...Priority -- the priority level for this attack
- ...SMessage -- the message to log when this attack happens
- ...Actions -- a list of actions to take when **RealSecure** sees this attack.

Also, there are tunable parameters for some of the checks. To prevent false alarms, set the sensitivity of these checks. **RealSecure**'s default configuration has reasonable values for these settings. These settings may vary, depending on the network. They are configurable by selecting the check in the Configure Signatures window and clicking on the **Advanced** button.



Time (in seconds) to wait before nuking unacknowleged connection reques Going too low with this time might kill off valid connections which are just taking a while to connect. Going too high might allow a SYN flood attack to keep the network offline. Anywhere from 6 to 12 seconds is a reasonable value.

Advanced Pattern Configuration window

Checks

☑ IP Fragmentation

This check probes for IP fragments with a size less than or equal to **IPFragThreshold**. Fragmentation requires that the data portion of the generated fragments (that is, everything excluding the IP header) be a multiple of 8 bytes for all fragments other than the final one. To check for an offset less than 16, use a setting of **2**. There is no need to change the default value.

☑ Ping Flooding

This check determines if more than **PingFloodPackets** are received in **PingFloodDelta** seconds. The default setting is 50 packets in 3 seconds. If the network is on a slow connection such as 14.4 PPP, consider making this setting more sensitive. Otherwise, the default value should suffice.

☑ **Arp Check**

If someone attempts to contact a host that is powered down, multiple address request packets are sent with no response. **ArpMaxUnAcked** sets how many requests are sent to an unresponsive host, before triggering an alarm.

☑ **Synflood Check**

A SYN flood is a Denial of Service attack created by filling up the listen queue of a machine so that there is no room for legitimate users to establish a connection. If this attack is detected, and the "Kill" action is set, **RealSecure** will implement a random drop algorithm that frees up an entry in the listen queue for a legitimate connection.

In order to optimize the effectiveness of this algorithm, it is necessary to set the advanced parameter 'SYNFloodHighWaterMark'. This is the number of SYNs to allow to wait in each machine's queue for a response before the random drop algorithm is implemented. This number should be smaller than the size of the listen queue for your machines by some percentage. A guideline for this is 70% of the size of your listen queue, but this value will need to be fine-tuned to find what works best for each individual network.

The larger the size of the listen queue, the more effective **RealSecure** will be in allowing the queue to remain open at most points in time. Contact your vendor for information about how to increase the size of this queue, or to determine its current value.

Sample Configurations

The **RealSecure** sample directory contains several example configuration files that demonstrate various ways to use **RealSecure** on a network. Modification of these samples is not necessary. Next, set aside time to preview **RealSecure**'s various features. Once each distinct option has been explored, consider then creating a customized written configuration (refer to the files in the rs/sample directory for more details regarding customizing a configuration).

Each sample configuration contains two files, `filter.cfg` and `features.cfg`. The `filter.cfg` file determines which packets on the Network, **RealSecure** is to examine and suitable action(s) based on those packets. The `features.cfg` file contains information about how to analyze the traffic being examined in the filter configuration.

To use these samples, copy the sample `filter.cfg` and `features.cfg` to the **RealSecure** GUI directory.

Sample Configuration 1: WebWatcher

```
web-features.cfg

web-filter.cfg
```

The WebWatcher configuration logs all URLs transmitted across a network to the **RealSecure** GUI and to an http.log file, while ignoring all other traffic. This allows for tracking and monitoring Web site traffic in real-time and stores Web page(s) logs for which user's access. The administrator should refer to the network security policy that limits the types of Web page(s) and sites' users are permitted to access. As well, consider restricting the hours in a day and allotted online time users are permitted to "Surf the Web."

This configuration allows for tracking and monitoring of how each user is utilizing the World

Wide Web (WWW) within an organization; all the while, not compromising speed or accessibility to the WWW. Identify violations to the network security policy either in real-time from the GUI or later by reviewing log files. Subsequently, take administrative action(s) based on the source and time of any such violation. Log source and time reports if necessary and use them to document network security violations as required.

This configuration produces logs that monitor Web traffic on the network Web server, without ever modifying it. Web traffic logs provide essential information that consequently administers a backup mechanism. These traffic logs both maintain and log all Web connections on the server; even if network disk space becomes full or logs become inadvertently deleted.

In the `filter.cfg` file, the source and destination addresses are set to the wildcard address, **0.0.0.0/0**. This file logs incoming connections from any source address to any destination address. If there is a specific machine installed to monitor outgoing Web connections, consider changing the source address value to the IP address of the machine; and only connections originating from that machine will be logged.

Ignore all outgoing traffic while monitoring the Web server. Modify the destination address to the IP address to that of the Web server. The source port in the configuration is set to 0, indicating connections coming from any source port will be logged. The Message value is set to Web. Priority is set to 2 (Medium). Connection establishment on the GUI is viewable real-time, since the Display action is enabled.

In the features.cfg file, the HTTPGetDecodeCheck is enabled, and all other checks disabled. The action for this check is set to Display and to log to a file.

Sample Configuration 2: Exploit Finder

exp-features.cfg

exp-filter.cfg

This configuration displays all attempts of known exploits against any machine on the network, or originating from any machine on the network. This immediately allows an administrator to check for a break-in, using one of the many ways **RealSecure** checks. **RealSecure** also detects if anyone inside the network is attempting to break-into other machines on the Internet.

To configure this, set up the features configuration to enable all of the exploit checks, while disabling all the simple informational decodes. The filter configuration is then set to filter out services the exploits are attempted against, in this case HTTP, SMTP, FTP, Rlogin, and Ident. Consider configuring the individual features to page or flag some other alert when detecting one of these exploits. These features will indicate a break-in attempt.

Sample Configuration 3: TCP Traffic

all-features.cfg

all-filter.cfg

This configuration displays all TCP traffic going through the network and displays all decodes and vulnerability attempts. This configuration is good for a small network having an active system administrator. Watching **RealSecure**, the administrator can monitor all of the TCP traffic on the network, users and passwords being attempted to each machine files being transferred over FTP and URLs being used over HTTP. These attempts are harder to detect on larger networks.

Changing the one line of the filter configuration to point to a specific machine, can significantly reduce the information that pinpoints all of the TCP traffic coming and going from one particular machine. If there is a suspected hacker on a machine, consider this option to monitor the traffic on the machine. If this is the case, consider enabling log in to a file in order to efficiently monitor the activities of the hacker.

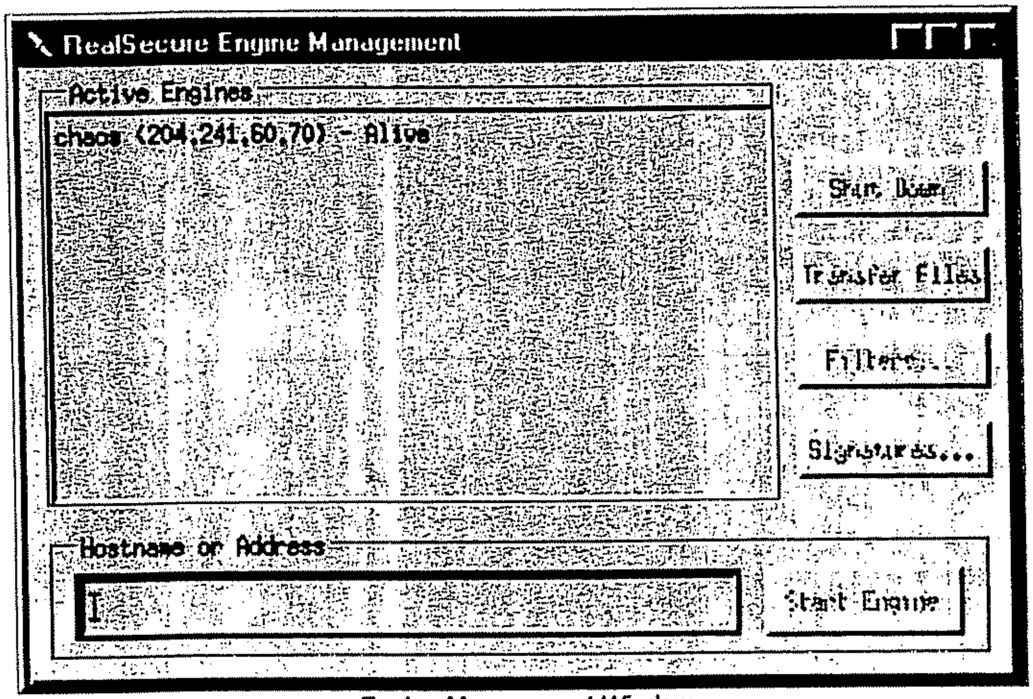
The filter configuration is defined to display all TCP traffic going through the network. Every feature is enabled in the features configuration.

Using RealSecure

Return to the Index

Chapter 4:
Using RealSecure™

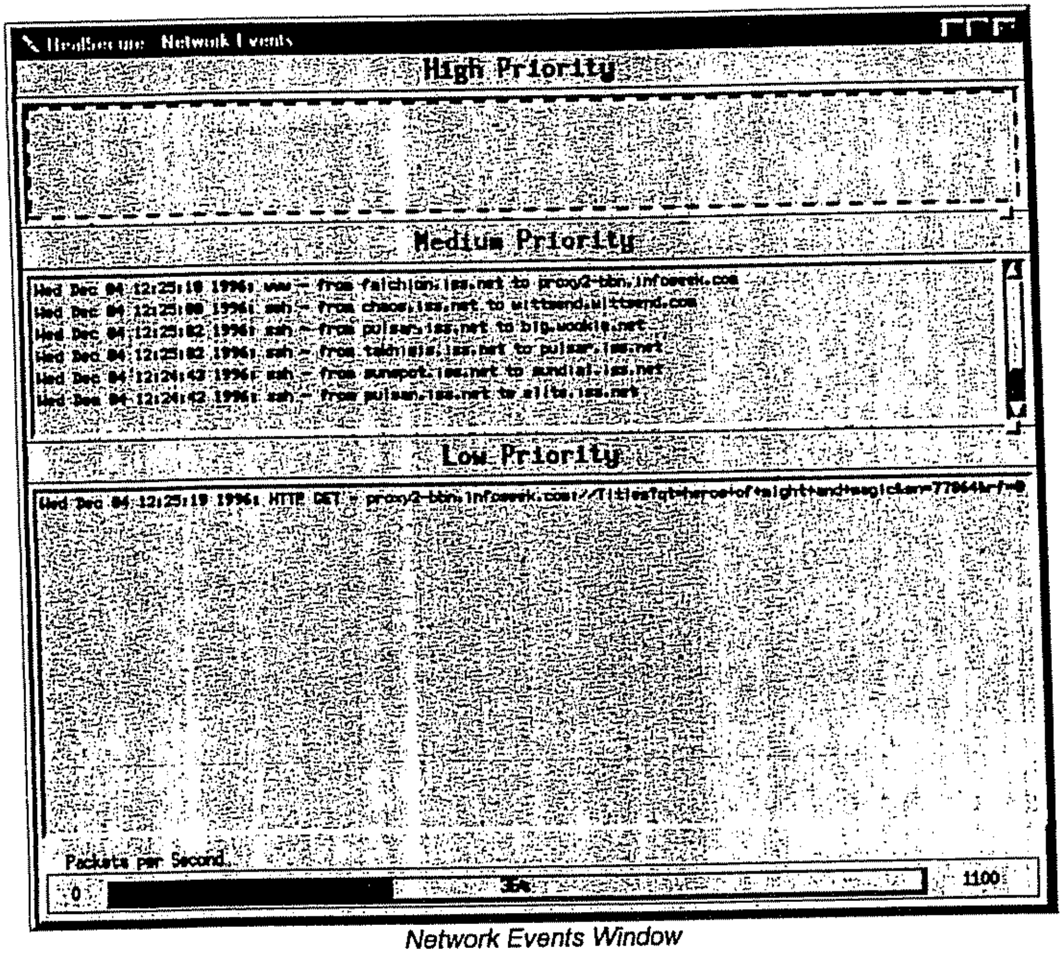
When selecting RealSecure from the RealSecure toolbar, the Engine Management window appears. This window allows for starting, stopping, and configuring engines on each host. If there are already engines running on the network, they will automatically contact the GUI and appear in this list within ten seconds.



chaos (204.241.60.70) - Alive

Engine Management Window

To start an engine, type the name or IP address of a host on which you wish to run an engine and press Enter, or click on **Start Engine**. After a short delay, the engine will appear in the engine list and the Network Events window will pop up to start displaying events.

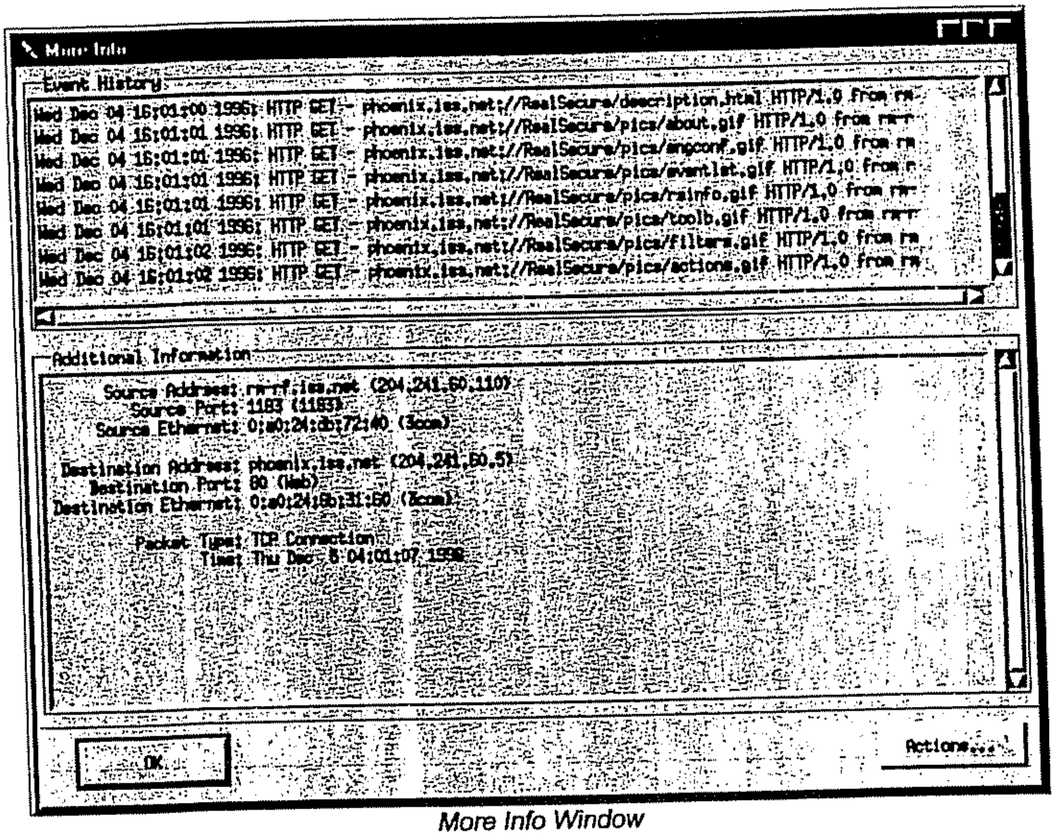


```
Wed Dec 04 12:25:19 1996: www - from falchion.iss.net to proxy2-bbn.infoseek.com
Wed Dec 04 12:25:08 1996: ssh - from chaos.iss.net to wittsend.wittsend.com
Wed Dec 04 12:25:02 1996: ssh - from pulsar.iss.net to big.wookie.net
Wed Dec 04 12:25:02 1996: ssh - from taichisis.iss.net to pulsar.iss.net
Wed Dec 04 12:24:43 1996: ssh - from sunspot.iss.net to sundial.iss.net
Wed Dec 04 12:24:42 1996: ssh - from pulsar.iss.net to elite.iss.net
```

```
Wed Dec 04 12:25:19 1996: HTTP GET - proxy2-bbn.infoseek.com://Titles?qt=heros+of+might+and+magic&an=77964&rf=0
```

Network Events Window

As events appear in the window, you can get more information about them by double-clicking on the entry.



```
Wed Dec 04 16:01:00 1996: HTTP GET - phoenix.iss.net://RealSecure/description.html HTTP/1.0 from rr
Wed Dec 04 16:01:01 1996: HTTP GET - phoenix.iss.net://RealSecure/pics/about.gif HTTP/1.0 from rr-r
Wed Dec 04 16:01:01 1996: HTTP GET - phoenix.iss.net://RealSecure/pics/engconf.gif HTTP/1.0 from rr
Wed Dec 04 16:01:01 1996: HTTP GET - phoenix.iss.net://RealSecure/pics/eventlst.gif HTTP/1.0 from r
Wed Dec 04 16:01:01 1996: HTTP GET - phoenix.iss.net://RealSecure/pics/rsinfo.gif HTTP/1.0 from rr-
Wed Dec 04 16:01:01 1996: HTTP GET - phoenix.iss.net://RealSecure/pics/toolb.gif HTTP/1.0 from rr-r
Wed Dec 04 16:01:02 1996: HTTP GET - phoenix.iss.net://RealSecure/pics/filters.gif HTTP/1.0 from rr
Wed Dec 04 16:01:02 1996: HTTP GET - phoenix.iss.net://RealSecure/pics/actions.gif HTTP/1.0 from rr
```

```
     Source Address: rr-rf.iss.net (204.241.60.110)
        Source Port: 1183 (1183)
    Source Ethernet: 0:a0:24:db:72:40 (3com)

Destination Address: phoenix.iss.net (204.241.60.5)
   Destination Port: 80 (Web)
Destination Ethernet: 0:a0:24:8b:31:60 (3com)

        Packet Type: TCP Connection
               Time: Thu Dec 5 04:01:07 1996
```

More Info Window

The More Info window displays information about the event, like the Ethernet addresses and TCP/UDP ports associated with the connection. Also, a list of all related events allows you to track a connection from start to finish. To close this window, click on OK. To perform some action with this connection, click on **Actions**. Note that certain actions are limited by the network state or the underlying protocol. For instance, you can't kill a connection that doesn't exist any more.

The bar at the bottom of the screen displays the number of packets per second seen by the engine(s). If they bar ever goes above 100%, that becomes the new maximum.

Events are added to the top of the list. Old events are moved toward the bottom. Each window (high, medium, and low) has a timeout value for its events. When the time expires for an event, the event is removed from the screen. If it was logged to a file, you still have a record of the event that will show up on any reports you generate. If the event occurs again after being removed, the new event will be displayed.

As you watch events occur, you may find that there are some which you do not wish to see. Go back to the configuration screens and set those events to be ignored. Alternatively, you may find that you need to see some events you have been ignoring. RealSecure is meant to change with your changing network needs. So feel free to modify or use the sample configurations. Make sure to keep a backup copy of the known good configurations.

Generating Reports

Return to the Index

Chapter 5:
Generating Reports™

After generating various logs of events, you will probably want to see a summary of network attacks. This can help you see where your major network problems are and modify security configurations appropriately. It is also useful for seeing trends in attacks and preparing a strategy for handling future intrusions.

This chapter covers the following topics:

➤ Playback Feature

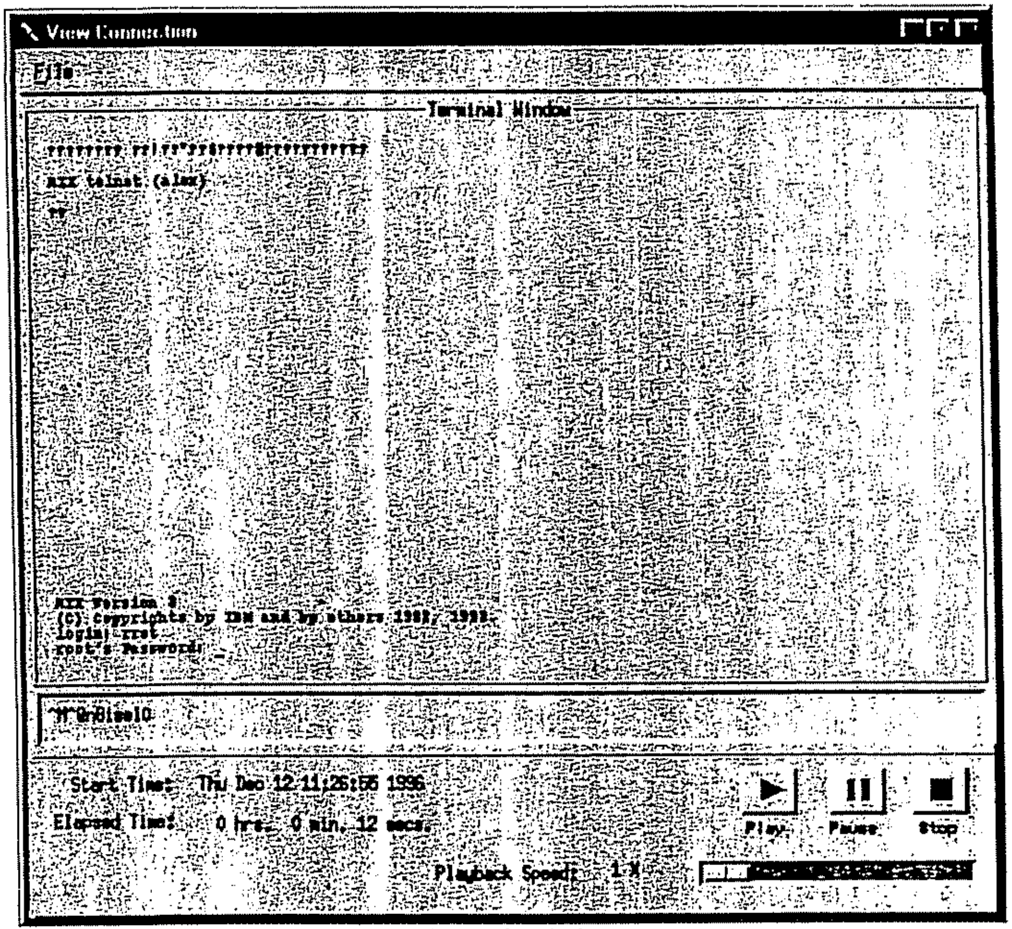
➤ Reporting Feature

Playback Feature

Although not a reporting feature *per se*, the **playback** feature of RealSecure allows it to show you what the intruder saw and typed. To use this feature, get a log generated with the **Log Raw** action and enter:

```
# playback <my-logged-raw-data-file>
```

A list of the connections recorded in that file is displayed.



```
????????? ??|??"???????@????????????
AIX telnet (alex)
??

AIX Version 3
(C) Copyrights by IBM and by others 1982, 1993.
login: root
root's Password: _
```

```
^M^@n@isel0
```

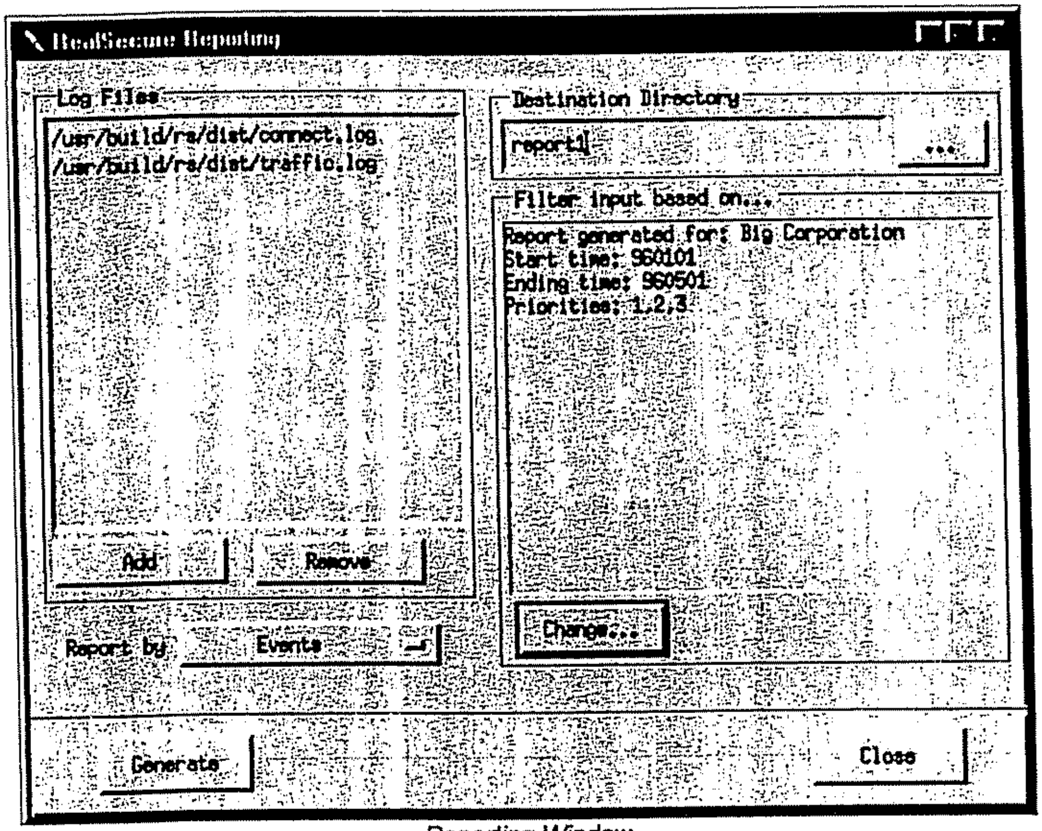
Start Time: Thu Dec 12 11:26:56 1996
Elapsed Time: 0 hrs. 0 min. 12 secs.
Playback Speed: 1 X

Playback Window

Select the one you wish to view and click the **Play** button. The play speed can be modified with the slider at the bottom of the screen. Pressing **Stop** gets you back to the list of connections.

Reporting Feature

The reporting feature uses logs generated with the **Log Info** action. To generate reports, click on the **Reports** button on the RealSecure toolbar. This displays the Reporting window.



Log Files:
/usr/build/rs/dist/connect.log
/usr/build/rs/dist/traffic.log

Destination Directory: report1

Filter input based on...
Report generated for: Big Corporation
Start time: 960101
Ending time: 960501
Priorities: 1,2,3

Report by: Events

Reporting Window

The Reporting window allows you to configure what log files are used to generate the report, where the report will be stored, and options to set for generating the report.

On the left of the screen is the list of log files to use to generate the report. To add another file to the list, click on the **Add** button and select the file you wish to use. To remove a file from the list, click on the name in the list and click on **Remove**. The **Report by** menu selects one of the three reports to generate: by source address, by destination address, and by event type (the default).

The destination directory for the reports can be changed by clicking on the ... button. If you select a directory that doesn't exist, RealSecure will create it for you.

The other list shows what parts of the logs will be used in reporting. For instance, you can generate a report on only certain priorities of events, certain time periods, and certain addresses. To modify this list, click on the **Change** button.

The **Change** button brings up a series of settings for report generation. The **Title** option allows you to specify a "Report Generated for..." title in the report. The **Source** and **Destination Addresses** fields allow you to specify an IP or range of IPs to limit the report. The **Start** and **End Times** fields allow you to specify a time period to report on, in yymmddhhmm.ss format (year, month, day, hour, minute, second). Here is one valid configuration:

> Report generated for: Internet Security Systems
>
> Start date: 9610250900

End date: 9611250900

Source addresses: 10.0.0.1,20.0.0.1-20.0.0.120

Destination addresses: Ken,Barbie

This generates a report on all events from Oct 25, 1996 at 9 am to Nov 25, 1996 at 9 am. It only shows events coming from the host 10.0.0.1 and hosts 20.0.0.1 to 20.0.0.120 that are destined for the two hosts Ken and Barbie.

After setting up your report, click on **Generate** to generate a report. After a short delay, a browser will pop up on the report title page. If not, you will receive an error message explaining why.

Features and Attack Signatures

Return to the Index

Appendix A:
Features and Attack Signatures

This appendix summarizes **RealSecure's** Features and the Attack Signatures it checks.

- IP Fragmentation
- Ping Flooding
- ARP Check
- IP Duplicate Check
- IP Half Scan
- IP Unknown Protocol
- UDP Bomb
- SYN Flood
- Source Routing
- Satan Vulnerability Check
- Chargen Denial of Service Vulnerability Check
- Echo Denial of Service Vulnerability Check
- TFTP Get Vulnerability Check
- TFTP Put Vulnerability Check
- Rwhod Vulnerability Check
- Finger User Decode
- Finger Bomb Vulnerability Check
- RTM Finger Vulnerability Check
- Rlogin Decoding
- Rlogin -froot Vulnerability Check
- FTP Username Decoding
- FTP Password Decoding
- FTP Site Command Decoding
- FTP GET File Decoding
- FTP PUT File Decoding
- FTP Mkdir Decoding
- FTP CWD ~root Vulnerability Check
- HTTP GET Decoding
- HTTP PHF Vulnerability Check
- HTTP Test-Cgi Vulnerability Check
- HTTP..Vulnerability Check
- HTTP Authentication Decode
- HTTP Java Decoding
- HTTP IIS 3.0 Asp Dot Vulnerability Check
- HTTP IIS 3.0 Asp 2E Vulnerability Check
- HTTP Internet Explorer 3.0 .URL/.LNK Vulnerability Check
- Ident User Decoding
- Ident Buffer Overflow Vulnerability Check
- Ident Newline Vulnerability Check
- POP Username Decoding

- POP Password Decoding
- RSH Decoding
- E-Mail From
- E-Mail To
- E-Mail Subject
- E-Mail VRFY
- E-Mail EXPN
- E-Mail WIZ Vulnerability Check
- E-Mail DEBUG Vulnerability Check
- E-Mail Pipe Vulnerability Check
- E-Mail Decode Vulnerability Check
- IRC Nick Decode
- IRC Channel Decode
- IRC Message Decode
- NNTP Username Decoding
- NNTP Password Decoding
- NNTP Group Decoding
- Talk Request Decoding
- Talk Flash Vulnerability Check
- HP/UX RemoteWatch Vulnerability Check

Features and Attack Signatures

☑ IP Fragmentation

An IP packet is sometimes split into several fragments when it is transmitted over the network. These fragments are then reassembled at the destination to form a full IP packet. Some routers that filter out packets based on information in the TCP header rely on the information in the first fragment, then blindly pass the remaining fragments. It is possible to construct individual fragments of an IP packet so that subsequent packets overlap. As a result, overwrite parts of the TCP header when they are reassembled at the destination. The result of this is that an intermediate filtering router is tricked into believing that a packet is destined for an allowed service. In reality, the packet is destined for a service that would normally be filtered.

☑ Ping Flooding

A Ping Flood is an attempt to saturate a network with packets in order to slow or stop legitimate traffic going through the network. A continuous series of ICMP Echo Requests are made to a target host on the network, which then responds with an ICMP Echo Reply. The continuing combination of requests and replies slow the network and cause legitimate traffic to continue at a significantly reduced speed or, in extreme cases, to disconnect.

☑ ARP Check

ARP, Address Resolution Protocol, is used to determine the Ethernet address of a machine on a network given its IP address. If an ARP is received for a machine on the network, it immediately sends a reply. If the machine the ARP is destined for has crashed or otherwise disconnected from the network, several ARPs will be sent to it without any response. This lack of response to ARP packets is used to determine if a machine on the network has crashed.

☑ IP Duplicate Check

Only one machine on a network should send packets with a specific IP address. If a second machine on the network starts to send packets claiming to have the same source address, a network problem has occurred. A machine on the network may be misconfigured to have the same IP address as another machine, causing network conflicts. The other possibility, is that a machine on the network may be sending out IP packets with a forged source address.

☑ IP Half Scan

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is waiting for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. If the destination host is not waiting for a connection on the specified port, it will respond with an RST packet instead of a SYN/ACK. Most system logs do not log completed connections until the final ACK packet is received from the source. Sending an RST packet instead of the final ACK results in the connection never actually being established; so no logging takes place. Because the source can identify whether the destination host sent a SYN/ACK or an RST, an attacker can determine exactly what ports are open for connections, without the destination ever being aware of probing.

☑ IP Unknown Protocol

A standard IP packet contains an 8-bit protocol field. Common values for this field include 6 (TCP), 17 (UDP), and 1 (ICMP). Attackers sometimes use a non-standard value for this field, in order to exchange data between machines without logging mechanisms detecting the data that is being transmitted.

☑ UDP Bomb

A UDP packet that is constructed with illegal values in certain fields will cause some older operating systems to crash when the packet is received. If the target machine does crash, it is often difficult to determine the cause. Most operating systems that are not vulnerable to this problem will silently discard the invalid packet, leaving no traces that it was being subjected to a malicious attack.

☑ SYN Flood

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is waiting for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. When the SYN/ACK is sent back to the source, a block of memory is allocated to hold information about the state of the connection that is currently being established. Until the final ACK is received or a timeout is reached, this block of memory sits unused, waiting for more information to be received from the source host. By sending numerous SYN packets to a host, the destination will exhaust the portion of memory it has on-hand to deal with opening connections. Legitimate connections will no longer be able to connect to the host. This situation

can be detected by the flood of SYN packets without accompanying responses. It can be corrected by sending the destination RST packets that correspond to the initial SYNs. This results in the destination host freeing up that block of memory and making room for a new legitimate connection.

☑ Source Routing

IP packets sent over the Internet are normally sent between different routers, in order to reach their final destination. The route each packet takes is determined dynamically by each router along the way. Enabling the source routing option on an IP packet allows the packet itself to make known to each router, the path it wishes to take to reach its final destination. By routing packets through a path that bypasses filtering routers and other normal security mechanisms, an attacker may be able to reach a host that normally could not be reached. Also, it can be used to authenticate an intruder to systems that rely on the source IP address for access control.

☑ Satan Vulnerability Check

(requires filter for UDP port 1)

This check will recognize if a Satan normal or heavy scan of a machine is taking place. Satan is a freely available tool that allows someone to scan a machine for services and a small set of common vulnerabilities.

☑ Chargen Denial of Service Vulnerability Check

(requires filter for UDP port 19)

This check will watch for attempts at performing a denial of service attack against a machine on the network by attempting to engage a machine in a chargen flood against itself.

☑ Echo Denial of Service Vulnerability Check

(requires filter for UDP port 7)

This check watches for attempts at performing a denial of service attack against a machine on the network by attempting to engage a machine in an echo flood against itself.

☑ TFTP Put Vulnerability Check

(requires filter for UDP port 69)

This check watches for attempts to transfer files to a machine using the Trival File Transfer Protocol (TFTP). This protocol can be used by attackers to transfer critical system files to a host that is being attacked.

☑ Rwhod Vulnerability Check

(requires filter for UDP port 513)

This check watches for a malformed rwho UDP packet containing a buffer overflow, that can be used by attackers to perform a denial of service attack against the rwho service or to attempt to execute arbitrary code on a remote machine.

☑ Rlogin Decoding

(requires filter for TCP port 513)

An Rlogin connection allows a user to remotely login to a host without a password by using a trust relationship between the account on the source machine and on the destination host. The source machine and username, along with the destination machine and username are logged with this feature.

☑ Rlogin -froot Vulnerability Check

(requires filter for TCP port 513)

If a remote user passes the name `-froot` to `rlogin` to a machine, certain operating systems will bypass normal security mechanisms and log in the user directly as `root`. This vulnerability allows anyone who can access the `rlogin` service on the target host to gain immediate `root` access to the machine.

☑ FTP Username Decoding

(requires filter for TCP port 21)

FTP, File Transfer Protocol, allows users to transfer files between machines. Username decoding discovers the name of the account being used to transfer files across the network.

☑ FTP Password Decoding

(requires filter for TCP port 21)

FTP passes a plain text password across the network in order to authenticate that a user has access to the files on the destination host. This password is discovered using FTP password decoding. This allows an administrator to log invalid password attempts, check passwords for strength against attack and keep complete logs of activity.

☑ FTP Site Command Decoding

(requires filter for TCP port 21)

The FTP site command allows a user to execute certain commands on a destination host in addition to the normal FTP facility of transferring files. In ordinary usage of FTP, this is not a commonly used command. While there may be a legitimate reason to execute site commands under certain circumstances, this facility has also been used to gain access. Consequently, an administrator may wish to view and log the site commands being executed to check for possible abuse.

☑ FTP GET File Decoding

(requires filter for TCP port 21)

Files being transferred from the destination host to the source host use a GET command in order to transfer the files. FTP GET decoding discovers all files that are being transferred to the source host over FTP.

☑ FTP PUT File Decoding

(requires filter for TCP port 21)

Files being transferred from the source host to the destination host use a PUT command in order to transfer the files. FTP PUT decoding discovers all files that are being transferred to the destination host over FTP.

☑ FTP Mkdir Decoding

(requires filter for TCP port 21)

FTP allows a user to create a new directory on the target machine. FTP Mkdir decoding discovers all new directories that are created through FTP.

☑ FTP CWD ~root Vulnerability Check

(requires filter for TCP port 21)

Certain versions of the FTP daemon allow access to files on a machine through a sequence of commands culminating with `CWD ~root`. This vulnerability allows an attacker who can access FTP on the target host to transfer files that he/she would not normally have access.

☑ HTTP GET Decoding

(requires filter for TCP port 80)

Pages, images, and all other information that is viewed through a Web browser on the World Wide Web are transferred through HTTP using the GET command. HTTP GET decoding discovers all Web pages being transmitted unsecurely to a machine. This allows an administrator to track, log and view Web traffic on the network.

☑ HTTP PHF Vulnerability Check

(requires filter for TCP port 80)

The `cgi-bin` script `PHF`, which comes preInstalled with several versions of NCSA and Apache Web servers, contains a vulnerability that allows anyone who can access a Web site to the machine(s).

☑ HTTP Test-Cgi Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the `cgi-bin` test-cgi script. This program, installed by default with certain versions of Apache and NCSA web servers, allows a remote attacker to gain information about the contents of the cgi-bin directory of the web server which can be used for further attacks.

☑ HTTP.. Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack to attempt to obtain information above the "ServerRoot" directory. Some web servers vulnerable to this attack will allow remote users to list the contents of any directory on the system using this type of attack.

☑ HTTP Authentication Decode

(requires filter for TCP port 80)

This decode will log the username and password that is being used to authenticate using HTTP Basic authentication to a web server. This authentication uses Base64 encoding and can be used for such purposes as determining what user accounts are logging into web servers from what machines, log brute-force attacks against the web server, and to keep general logs of username and password attempts.

☑ HTTP Java Decoding

(requires filter for TCP port 80)

This decoding recognizes when a web browser attempts to obtain a file containing Java bytecode. This should only occur if a user has Java enabled on their web browser.

☑ HTTP IIS 3.0 Asp Dot Vulnerability Check

(requires filter for TCP port 80)

Microsoft's IIS 3.0 server has a security hole that allows execution of code by inserting a '.' after an active server push URL. This check will recognize attempts to exploit this vulnerability.

☑ HTTP IIS 3.0 Asp 2E Vulnerability Check

(requires filter for TCP port 80)

Microsoft's IIS 3.0 server installed with the hot-fix to solve the ASP Dot vulnerability introduced a new security hole that allows viewing the contents of an active server push URL by using the hexadecimal value '2e' instead of a '.' in the URL name. This check will recognize attempts to exploit this vulnerability to view the contents of pages.

☑ HTTP Internet Explorer 3.0 .URL/.LNK Vulnerability Check

(requires filter for TCP port 80)

Microsoft's Internet Explorer versions 3.0 and 3.01 have a vulnerability which results in a web site being able to execute an arbitrary program on a machine running Microsoft Windows and browsing the web using MSIE. This vulnerability check will detect when a web site attempts to exploit this vulnerability.

☑ Ident User Decoding

(requires filter for TCP port 113)

The Ident port is used by services to identify the account by which a connection is being made on a machine. This can be used to track a connection back to a specific user on a multi-user machine.

☑ Ident Buffer Overflow Vulnerability Check

(requires filter for TCP port 113)

Certain programs that connect back to the `ident` service to log user information, expect a properly formatted response. If the response is longer than expected, the buffer that the response is read into is overflowed, allowing the remote user to execute commands on the host machine.

☑ Ident Newline Vulnerability Check

(requires filter for TCP port 113)

Certain programs that connect back to the `ident` service to log user information expect a properly formatted response. If the response contains newlines, the response may be improperly parsed, allowing the remote user to execute commands on the host machine.

☑ POP Username Decoding

(requires filter for TCP port 119)

The POP service is used by numerous e-mail programs to retrieve e-mail from a mail server and read it on a local machine. POP username decoding discovers the username of the user who is reading mail through the POP service.

☑ POP Password Decoding

(requires filter for TCP port 119)

POP password decoding discovers all successful and unsuccessful passwords that a user attempts to use to login to a mail server using POP.

☑ RSH Decoding

(requires filter for TCP port 512)

RSH, the remote shell command, allows a user to execute a shell command over the network using a trust relationship between the user on the local machine and the user account on the remote machine. RSH decoding discovers both the local and remote usernames as well as the command that is being executed.

☑ E-Mail From

(requires filter for TCP port 25)

This decoding discovers the sender of all mail that is sent over the network using SMTP.

☑ E-Mail To

(requires filter for TCP port 25)

This decoding discovers the recipient of all mail that is sent over the network using SMTP.

☑ E-Mail Subject

(requires filter for TCP port 25)

This decoding discovers the subject line of all mail that is sent over the network using SMTP.

☑ E-Mail VRFY

(requires filter for TCP port 25)

The VRFY command is used to verify if a user on a remote system exists. This is sometimes used legitimately to determine if the recipient of a message at the intended destination is able to receive the message. It is also sometimes used to gain information about users on a system by attempting to determine if certain common account names exist on a machine.

☑ E-Mail EXPN

(requires filter for TCP port 25)

The EXPN command is used to expand the address of a user on a remote system. This is sometimes used legitimately to determine the full address of an intended mail recipient. It is also sometimes used to gain information about users on a system by trying to find out if certain common account names exist on a machine.

☑ E-Mail WIZ Vulnerability Check

(requires filter for TCP port 25)

The WIZ command in Sendmail existed to allow access to a machine under certain circumstances. It is no longer present in current versions of Sendmail, but old versions still in use may allow an attacker to gain `root` access to a machine by using this command.

☑ E-Mail DEBUG Vulnerability Check

(requires filter for TCP port 25)

The DEBUG command in Sendmail existed to allow debugging of a remote Sendmail daemon. It is no longer present in current versions of Sendmail, but old versions still in use allow an attacker to gain `root` access to a machine by using this command remotely.

☑ E-Mail Pipe Vulnerability Check

(requires filter for TCP port 25)

By inserting a pipe (|) character into certain fields in an e-mail, Sendmail may be forced to execute a command on the remote host. This results in a remote attacker being able to execute commands as `root` on the machine.

☑ E-Mail Decode Vulnerability Check

(requires filter for TCP port 25)

By sending mail to decode or uudecode alias that is present in some systems, a remote attacker may be able to create or overwrite files on the remote host.

☑ TFTP Get Vulnerability Check

(requires filter for UDP port 69)

This check watches for attempts to transfer files from a machine using the Trivial File Transfer Protocol (TFTP). This protocol is sometimes legitimately used for bootstrapping by diskless workstations, but it is more often used by attackers to attempt to obtain a password file or other critical system files.

☑ Finger User Decode

(requires filter for TCP port 79)

This decode watches for finger attempts and reports the user (or all users if the attempt was aimed at the whole machine) that the finger was aimed at. Finger has a legitimate use, but is also often used by attackers to gain more information about a machine such as account names, real names, and trusted hosts.

☑ Finger Bomb Vulnerability Check

(requires filter for TCP port 79)

This check watches for attempts to perform a denial of service attack against a machine or for redirecting finger attempts across machines. Redirecting finger attempts is often used by an attacker to hide the original source address of a finger attempt.

☑ RTM Finger Vulnerability Check

(requires filter for TCP port 79)

This check watches for a buffer overflow attempt on the finger service that is used by attackers to attempt to gain access to a machine remotely. This vulnerability is named for Robert T. Morris, author of the Internet Worm that originally popularized this vulnerability.

☑ IRC Nick Decode

(requires filter for TCP port 6667)

This decode watches for changes of a user's nickname on Internet Relay Chat.

☑ IRC Channel Decode

(requires filter for TCP port 6667)

This decode watches for channels that are joined by a user on Internet Relay Chat.

☑ IRC Message Decode

(requires filter for TCP port 6667)

This decode watches for messages that are sent out by a user on Internet Relay Chat.

☑ NNTP Username Decoding

(requires filter for TCP port 119)

The NNTP service is used to read, post, and exchange news from a news server. NNTP user decoding discovers the username of the user who is reading or posting news through the NNTP service.

☑ NNTP Password Decoding

(requires filter for TCP port 119)

The NNTP service is used to read, post, and exchange news from a news server. NNTP password decoding discovers the password attempted to login to the news server in order to read or post news.

☑ NNTP Group Decoding

(requires filter for TCP port 119)

The NNTP service is used to read, post, and exchange news from a news server. NNTP group decoding

- discovers the name of the newsgroup that a user is accessing on the news server.

☑ Talk Request Decoding

(requires filter for UDP port 517 and 518)

The Talk service is used to engage in a real-time chat with a user on a remote machine. Talk Request decoding discovers the name and machine that a talk request is being sent to, along with the name and machine of the person who is originating the talk request.

☑ Talk Flash Vulnerability Check

(requires filter for UDP port 517 and 518)

The talk service allows the user originating a talk request to specify an arbitrary string to display for the origin of the talk request. If this string contains a particular escape sequence, it is possible to cause a temporary denial of service attack by mangling the contents of a user's screen. This is commonly known as 'flashing' a user.

☑ HP/UX RemoteWatch Vulnerability Check

(requires filter for TCP port 5556)

Certain versions of HP/UX that come with the RemoteWatch package installed have a vulnerability which allows a remote attacker to execute arbitrary commands through the RemoteWatch service on the target machine. This vulnerability check will watch accesses to the RemoteWatch service and determine if these accesses are attempting to exploit his vulnerability.

Return to the Index